```
/*
 * berra.c
 *
 * Created: 2017-04-07 10:06:40
 * Author : ine14ssu
 */
/*
 * plantpuppy.c
 *
 * Created: 2017-03-28 14:53:54
 * Author : ine14ssu
 */

#include <avr/io.h>
#include <util/delay.h>
#define F_CPU 1000000UL //1MHz
#include <avr/interrupt.h>

//Avbrott
#define Dataport PINA
#define ButtonUp (1<<PA5)
#define ButtonDown (1<<PA6)
#define ButtonLow (1<<PA4)
#define ButtonMid (1<<PA3)
#define ButtonHigh (1<<PA7)
#define reset



/*Deklaration av metoder*/

void init();
void watering();
void increaseLevel();
void decreaseLevel();
void writeStatus();
void openVent();
void closeVent();
void watering();
void writeLCD(char *str);
void newLCDLine();
void writeLCDMessage(char val);
void startLCD();
void clearLCD();
void initialModeLCD();
void startLed();
```

```
void Ledoff();
void initADC();
void stopLCD();
void initButtonInterrupt();


/*Variabeldeklaration*/
int delayLCD = 2;
char btnLow;
char btnMid;
char btnHigh;
char btnUp;
char btnDown;
int humidity;
unsigned short int button;
int waterAmount = 10; //1 , 100 , 1000
int humidLevel = 50;  // 0 till 100

int val;

/*
Avbrottsdelen
- Hit går den när det kommer en extern input från INT0
- sen går den tillbaka dit den började
**/

ISR(INT0_vect){

if((ButtonUp & PINA)!=0){
button=ButtonUp;
}
if((ButtonDown & PINA)!=0){
button=ButtonDown;
}
if((ButtonLow & PINA)!=0){
button=ButtonLow;
}
if((ButtonMid & PINA)!=0){
button=ButtonMid;
}
if((ButtonHigh & PINA)!=0){
button=ButtonHigh;
}
}

/*Initial bestämmelse av knapparnas värden*/
```

```
void init(){

	btnDown = (0b01000000 & PINA);
	btnUp = (0b00100000 & PINA);
	btnLow = (0b00010000 & PINA);
	btnMid = (0b00001000 & PINA);
	btnHigh = (0b10000000 & PINA);
	DDRA=0b11111001;
	DDRD = 0b00100100;
	sei();
}

void initADC(){
	ADMUX = 0b11100001;
	ADCSRA = 0b11101010;

}

ISR(ADC_vect) {
humidity=ADCH;

}

void initButtonInterrupt(){
	MCUCR = 0b00000011;
	GICR = 0b01000000;
}
void increaseLevel(){
	if(humidLevel > 200){
		return;
	}
	humidLevel = humidLevel + 20;
	writeStatus();
	return;
}
void decreaseLevel(){
	if(humidLevel < 0){
		return;
	}
	humidLevel = humidLevel - 20;
	writeStatus();
	return;
}
void writeStatus(){
		clearLCD();
```

```
                char fukt[5];
                sprintf(fukt,"%d",(int)humidLevel);
                writeLCD("Humidity ");
                writeLCD(fukt);
                clearLCD();
}
void openVent(){
        DDRD |= (1<<5);
        PORTD |=(1<<5);
}
void closeVent(){
        PORTD &= ~(1<<5);
}
void watering(){
        openVent();
        writeLCD("Water");
        _delay_ms(val);
        clearLCD();
        closeVent();
        _delay_ms(100);
}
void writeLCD(char *str){
        int i;
        for(i=0;str[i]!='\0';i++){
                if(i>31){
                        return;
                }
                writeLCDMessage(str[i]);
                _delay_ms(delayLCD);
        }
        return;
}
void newLCDLine(){
        PORTB=0b00001000;
        _delay_ms(delayLCD);
        portRWone();
        _delay_ms(delayLCD);
        portEone();
        _delay_ms(delayLCD);
        portRWzero();
        _delay_ms(delayLCD);
        portEzero();
        _delay_ms(delayLCD);
        portEone();
        _delay_ms(delayLCD);
        portRWone();
```

```
	_delay_ms(delayLCD);
	portRSzero();

}
void writeLCDMessage(char val){

	portEone();
	if(val == 'Ö'){
		PORTB=0b11101111;
	}else{
		_delay_ms(delayLCD);
		PORTB=val;
		}
		portRSone();
		portRWzero();
		portEzero();
		portEone();
		portRWone();
		portRSzero();

}
void clearLCD(){

	portRSzero();
	portRWzero();
	PORTB=0b00000001;
	_delay_ms(delayLCD);

	portEone();
	portEzero();
	portEone();

}
void stopLCD(){
	PORTB = 0b00001110;
	_delay_ms(delayLCD);
	portEone();
	portEzero();
}
void startLCD(){
	PORTB=0b00001111;
	_delay_ms(delayLCD);
	portEone();
	portEzero();
}
void initialModeLCD() {
```

```
        clearLCD();
        DDRB=0b11111111;
        _delay_ms(delayLCD);
        DDRD=0b00010011;
        _delay_ms(delayLCD);
        portEone();
}
void portRWzero(){
        PORTD &= ~(1<<1);
}
void portRWone(){
        PORTD |= (1<<1);
}
void portEzero(){
        PORTD &= ~(1<<4);
}
void portEone(){
        PORTD |=(1<<4);
}
void portRSzero(){
        PORTD &= ~(1<<0);
}
void portRSone(){
        PORTD |= (1<<0);
}
void startLed(){
PORTA |= (1<<2);
}
void Ledoff(){
 PORTA &= ~ (1<<2);
}
void displayAmount() {
        clearLCD();
        char water[5];
        sprintf(water,"%d",(int)waterAmount);
        writeLCD("Water amount ");
        writeLCD(water);
        _delay_ms(100);
        clearLCD();
        }
int main(void)
{

        init();

        initialModeLCD();
```

```
    startLCD();
    clearLCD();

    writeLCD("Welcome!");
    _delay_ms(1000);
    clearLCD();
    writeLCD("Choose");
    _delay_ms(1000);
    clearLCD();
    writeLCD("water amount");
    _delay_ms(2000);
    clearLCD();
    writeLCD("Press button");
    _delay_ms(1000);
    clearLCD();
    writeLCD("Low Mid High");
    _delay_ms(2000);
    clearLCD();
    writeLCD("Choose");
    _delay_ms(1000);
    clearLCD();
    writeLCD("humidity level");
    _delay_ms(2000);
    clearLCD();
    writeLCD("press button");
    _delay_ms(1000);
    clearLCD();
    writeLCD("up or down");
    _delay_ms(2000);
    clearLCD();

    initButtonInterrupt();
    clearLCD();
    initADC();
    clearLCD();

    char hej[6];
    sprintf(hej,"%d",(int)humidity);
    writeLCD("Humidity in");
    clearLCD();
    writeLCD("soil is ");
    writeLCD(hej);
    clearLCD();

while (1)
{
```

```
                if(button!=0){
                        startLed();
                switch(button){
                case ButtonUp: increaseLevel();  break;
                case ButtonDown: decreaseLevel();  break;
                case ButtonLow: waterAmount=1; displayAmount();  break;
                case ButtonMid: waterAmount=10; displayAmount(); break;
                case ButtonHigh: waterAmount=100; displayAmount();  break;
                case 0: break;
                }
                Ledoff();
                button=0;
                }
                if (humidity < humidLevel){
                        val = waterAmount*10;
                        watering();
                }
    }
}
```